4BA
ANALIZA BIZNESOWA

USER STORY

Krótkie podsumowanie

Autorzy:
Paulina Grębska i Radosław Grębski

Partnerzy:

Czym jest User Story?

User Story, czyli Historyjka Użytkownika, to pojęcie jednoznacznie związane ze zwinnymi metodykami wytwarzania oprogramowania. Gdy myślimy Scrum, zwykle na myśl przychodzi nam User Story i na odwrót.

Czym zatem jest User Story? **Jest to popularna i szeroko stosowana technika dokumentowania wymagań.** Opisuje w krótkiej i zwięzłej formie to, co dany użytkownik (interesariusz) chce zrobić lub czego potrzebuje i po co jest mu to potrzebne. Skupia się zatem na dostarczanej wartości.

Elementy User Story

Statement of Value

Bywa, że jako User Story, rozumiemy jedynie jego zasadniczą część, czyli tzw. "Statement of Value". **Jest to zdanie opisujące jaka wartość jest dostarczana, gdy interesariuszowi zapewniona zostanie określona funkcjonalność lub jakość.**

Zwykle "Statement of Value" opiera się na jednym z kilku szablonów. Chyba najpopularniejszy z nich to:

"As a <who?> I need to <what?>, so that <why?>."

Po polsku brzmi to mniej więcej tak:

"Jako <kto?> chcę <co?>, aby <po co?>"

Mamy zatem konstrukcję składającą się z 3 elementów:

AKTOR <kto?> – wskazuje konkretnego interesariusza (lub konkretny typ wchodzącego w interakcje z systemem, np. Administrator, Klient, Dostawca etc.). Zwykle wyróżnić można różne grupy użytkowników korzystających z systemu w odrębnych celach, posiadających zazwyczaj odrębne uprawnienia.

CZYNNOŚĆ <co?> – opis tego, co dany aktor zamierza zrobić w systemie. Opis wykonywanej przez niego czynności. Można w tej części opisać również jakość, która ma być dostarczona, a więc skupić się na wymaganiach niefunkcjonalnych.

WARTOŚĆ <po co?> – ta część określa właściwie sens implementacji, gdyż mówi o dostarczanej wartości. Niestety często ten właśnie element nie jest prawidłowo opisywany lub jest pomijany. Stanowi to poważny problem, gdyż tracimy z oczu wartość biznesową, która jest dostarczana, a to przecież ona jest najważniejsza.

Tytuł

Traktowany jest jako opcjonalny. Zwykle krótko, w formie kilku wyrazów, wyraża cel danego User Story.

Konwersacja

User Story należy traktować jedynie jako wstęp do dalszej dyskusji z zespołem projektowym i innymi interesariuszami oraz dalszych rozważań i związanego z nimi modelowania. Konwersacji, jako elementu User Story, nie należy traktować jako widoczny, zapisany fragment tekstu. To raczej podkreślenie faktu, że każda Historyjka Użytkownika wymaga odpowiedniego przedyskutowania z interesariuszami, a w szczególności z zespołem deweloperskim.

Kryteria akceptacji

Kryteria akceptacji pomagają uszczegółowić User Story oraz pomóc wyznaczyć jego zakres. Stanowią listę warunków, jakie muszą być spełnione, aby można było uznać, że zakładana wartość została dostarczona.

Czy User Story nadaje się do wszystkiego?

User Story jako narzędzie daje możliwość krótkiego i zwięzłego opisu potrzeby konkretnego interesariusza, którą możemy dodatkowo doprecyzować stosując kryteria akceptacji, a następnie poddać dalszej dyskusji i rozważaniom w celu odpowiedniego rozwiązania. Brzmi dobrze, na tyle dobrze, że stosowanie tej techniki bywa często nadużywane.

Spotkałem się kiedyś z opinią, że Scrum nie uznaje dokumentacji, a jeśli już to jedyną dopuszczalną jej formą są właśnie User Story. Oczywiście jest to błędne podejście.

Sytuacja taka miała miejsce, gdy system, który tworzył mój zespół, miał się integrować z systemem rozwijanym przez inny zespół. Poprosiłem wtedy o przekazanie mi dokumentacji systemu, w celu zapoznania się z jego funkcjonalnością, udostępnianym interfejsem etc. W odpowiedzi otrzymałem listę 153 User Stories dotyczących rozwijanego systemu... Oczywiście powstały one na przestrzeni kilku lat i te które powstały później modyfikowały funkcjonalność opisaną we wcześniejszych.

Stworzenie spójnej wizji systemu na podstawie przejrzenia przesłanej listy wydawało się zadaniem wybitnie trudnym. Jest to dobry przykład na to, iż User Stories, choć przydatne, nie nadają się do wszystkiego.

Do czego User Stories się NADAJĄ?

Nadają się do krótkoterminowego dokumentowania wymagań ze zwróceniem szczególnej uwagi na dostarczaną wartość. Ponadto służą jako wstęp do dyskusji mającej na celu wypracowanie wspólnego zrozumienia i dalszą pracę nad rozwiązaniem.

Co więcej, ze względu na swoją naturę pozwalającą na podział pracy na małe i zwięzłe funkcjonalności lub wymagania jakościowe, mogą być wykorzystywane do:

- → Dyskutowania priorytetów,
- → Szacowania pracochłonności,
- → Planowania rozwoju produktu i jego wersji,
- → Mapowania wysokopoziomowych wymagań,
- → Projektowania testów akceptacyjnych,
- → Śledzenia i raportowania postępów pracy.

Do czego User Stories się NIE NADAJĄ?

Przede wszystkim nie nadają się do długoterminowego dokumentowania wymagań, a także do tworzenia przy ich pomocy dokumentacji służącej do rozwoju i utrzymania systemu. Głównie ze względu na to, że na ich podstawie, bez odpowiedniej dokumentacji wspomagającej, ciężko jest zrozumieć pełny i aktualny obraz systemu.

Należy również pamiętać, że samo User Story, bez towarzyszącego mu kontekstu, to zwykle zbyt mało dla zespołu deweloperskiego, by mógł efektywnie pracować. Dla pełnego i poprawnego zrozumienia wprowadzanych zmian, ich celu i wpływu na poszczególne komponenty systemu, zespół musi mieć szerszy kontekst. **Można go zapewnić stosując dodatkowo inne techniki i formy dokumentacji.**

Cechy poprawnego User Story

Często można się spotkać z User Stories pisanymi na szybko, bez dokładnego przemyślenia, byle tylko spełnić wymagania wynikające z przyjętego szablonu. Oczywiście ma to katastrofalne skutki. Jeśli jedynym celem autora jest wpasowanie się w szablon bo "tak kazali" lub " taki mamy standard", to oznaczy, że nie warto zadawać sobie trudu i lepiej opisać wymagania w inny sposób, chociażby używając języka naturalnego.

Jak zatem pisać User Story, aby było to efektywne i użyteczne? Warto zwrócić szczególną uwagę na ich jakość. Aby to ułatwić, powstał pewien użyteczny skrót, tworzący łatwe do zapamiętania, angielskie słowo INVEST.

I *jak* ***Independent***

User Stories powinny być od siebie niezależne. Oznacza to, że mogą być zrozumiane, zaimplementowane oraz mogą dostarczać wartości, niezależnie od siebie.

Oczywiście nie oznacza to braku jakichkolwiek relacji. Tak jak pomiędzy wymaganiami opisanymi w dowolny sposób, tak również pomiędzy User Stories mogą pojawiać się relacje. Mogą one wskazywać, że dana Historyjka Użytkownika musi być zaimplementowana przed inną lub zaimplementowanie jednej, znacząco zmniejszy wysiłek zaimplementowania innej. Celem jest natomiast to, by User Story, stanowiło pewną, odrębną i logiczną całość.

Analogią może być budowa domu i wylewanie fundamentów oraz stawianie ścian. Oczywiście jest między nimi pewna relacja i jedna czynność musi być wykonana przed drugą, jednak stanowią osobne i logicznie odrębne etapy budowy.

Antyprzykładem byłoby przedstawienie w ramach oddzielnych historyjek montażu kaloryferów oraz rur centralnego ogrzewania. Elementy te zdecydowanie samodzielnie nie dostarczają wartości.

Przykłady niepoprawnego Statement of Value

Jako klient chcę mieć przygotowaną bazę danych, aby w przyszłości, dane zamówienia mogły być w niej zapisane.

Jako klient chcę mieć przetestowaną funkcjonalność zamawiania produktów, aby uniknąć problemów z ich zamówieniem.

Częste błędy:

- Tworzenie oddzielnych User Stories dotyczących technicznych elementów rozwiązania (np. oddzielne opisane przygotowanie schematu danych, opracowanie logiki etc.)
- Tworzenie oddzielnych User Stories dotyczących etapów procesu tworzenia oprogramowania (np. oddzielnie analiza, tworzenie kodu, testowanie etc.)

N *jak* **Negotiable**

User Story powinno być negocjowalne, czyli pozostawiać przestrzeń na dyskusje i negocjacje co do sposobu dostarczenia rozwiązania. Niepoprawnym jest tworzenie zbyt szczegółowych i "technicznych" User Story narzucających zespołom deweloperskim dokładny sposób tworzenia rozwiązania. Zabija to kreatywność i naraża na ryzyko pominięcia potencjalnie lepszego rozwiązania.

Przykłady niepoprawnego Statement of Value

Jako klient chcę wpisać w oddzielnych polach ulicę, numer budynku, numer mieszkania, miejscowość, kod pocztowy i ulicę oraz potwierdzić zapisanie adresu, klikając na przycisk "Zapisz adres", aby moje dane zostały zapisane w bazie danych.

Jako klient chcę otrzymywać mailem dokument w PDF zawierający numer katalogowy artykułu, jego cenę i nazwę, aby uzyskać trwałe potwierdzenie dokonanej transakcji.

Częste błędy:

- Definiowanie zbyt szczegółowego rozwiązania w treści User Story
- Brak pozostawienia jakiejkolwiek przestrzeni na kreatywność zespołu

V *jak* **Valuable**

Poprawne User Story wyraża wartość, która będzie dostarczona poprzez jego zaimplementowanie. Sama jego konstrukcja narzuca, iż powinniśmy wskazać: Kto? Co? Po co? i właśnie "Po co?" jest szczególnie ważne, gdyż wyraża sedno potrzeby biznesowej i dostarczanej wartości. Jeśli implementacja Historyjki Użytkownika nie dostarcza żadnej wartości, to po co się nią zajmować? Oznacza to jedynie poniesienie kosztu implementacji, bez wyraźnego zysku, a więc marnotrawstwo.

Przykłady niepoprawnego Statement of Value

Jako klient chcę mieć zapisać standardowy adres dostawy.
Jako pracownik chcę mieć możliwość przeglądania zamówień.

Częste błędy:

- Brak wskazania wartości dostarczanej interesariuszom

E *jak* ***Estimable***

Poprawnie określone User Story musi być możliwe do "oszacowania", czyli określenia wysiłku niezbędnego do jego zaimplementowania. Nie chodzi tu oczywiście o dokładne wyceny w dniach lub co gorsza godzinach, a o estymacje relatywne, wykorzystujące relatywne jednostki (jak np. Story Pointy) i bazujące na wcześniejszych doświadczeniach zespołu deweloperskiego. Estymację znacząco ułatwia zdefiniowanie jasnych i jednoznacznych kryteriów akceptacji.

Przykłady niepoprawnego Statement of Value

Częste błędy:

- Brak jasnego zakomunikowania celu biznesowego i zakresu User Story
- Brak omówienia User Story z zespołem deweloperskim

S *jak* ***Sized Appropriately***

Właściwie to im "mniejsze" User Story, tym lepiej. Oczywiście w granicach rozsądku. Granicą sensownej wielkości (wynikającej z relatywnej wyceny) jest możliwość dostarczenia w ramach jednej iteracji (Sprintu). Jeśli z estymacji jasno wynika, że Historyjki Użytkownika nie uda się dostarczyć w ramach iteracji, należy ją podzielić na mniejsze elementy.

Przykłady niepoprawnego Statement of Value

Częste błędy:

- Definiowanie User Story o zbyt dużym zakresie, niemożliwych do dostarczenia w ramach jednej iteracji

T *jak* ***Testable***

Tak jak każde wymaganie, również wymaganie opisane w formie User Story musi być testowalne. Oznacza to mniej więcej tyle, że musi istnieć możliwość obiektywnego sprawdzenia, czy zostało zaimplementowane i dostarczyło zakładaną wartość, określonemu interesariuszowi. Innymi słowy, musi być możliwy do zdefiniowania i przeprowadzenia test, który to weryfikuje w jednoznaczny sposób.

Przykłady niepoprawnego Statement of Value

Jako użytkownik chcę szybko wygenerować raport, aby cały proces zajmował mało czasu.
Jako klient chcę, aby wszystkie błędy zostały usunięte z systemu.

Częste błędy:

- Niejasne i niejednoznaczne (a przez to nieweryfikowalne) wymagania
- Brak lub niejasne kryteria akceptacji

Jak podzielić zbyt duże User Story?

Jedną z cech poprawnego User Story jest "Sized Appropriately". Granicą wielkości User Story jest możliwość implementacji w ramach jednej iteracji. Jeśli nie jest to możliwe, mamy do czynienia z tzw. Epikiem (ang. Epic) i należy go podzielić.

Najważniejszą zasadą jest to, aby po podziale, nowo powstałe User Stories niezależnie dostarczały wartości interesariuszom. Oczywiście warto zwrócić również uwagę na pozostałe cechy tworzące akronim INVEST. Wskazane jest aby dzielić Historyjki Użytkownika wertykalnie, tzn. dzielić je względem funkcjonalności. Innymi słowy tak, aby każde z nich dało się zaprezentować niezależnie finalnym użytkownikom.

Podział horyzontalny, wynikający z kolejnych kroków implementacji, np. przygotowanie struktury danych, implementacja logiki biznesowej i opracowanie interfejsu użytkownika, w ramach oddzielnych historyjek, jest absolutnie niewskazany. Jeszcze gorszym pomysłem jest podział User Story względem kolejnych faz procesu tworzenia oprogramowania, czyli np. projektowanie, implementacja i testowanie opisane w ramach oddzielnych Historyjek Użytkownika.

Poniżej prezentuję kilka sposobów podziału zbyt dużych (niemożliwych do zaimplementowania w jednej iteracji) User Story. Oczywiście sposobów jest więcej, a jedynym ograniczeniem jest tutaj kreatywność.

Według kroków procesu

Jeśli proces składa się z wielu kroków, każdy z nich możemy opisać za pomocą oddzielnej Historyjki Użytkownika. Banalnym przykładem może być proces składania zamówienia w sklepie internetowym. Może się on składać np. z dodawania towaru do koszyka, wyboru metody płatności, wyboru metody dostawy i podsumowania oraz potwierdzenia zamówienia. Cały proces może być opisany jako ogólna Historyjka użytkownika.

Przykład

Jako klient chcę złożyć zamówienie w sklepie internetowym, aby otrzymać wybrane przeze mnie produkty.

- Istnieje możliwość przeglądania produktów dostępnych w sklepie
- Istnieje możliwość wskazania produktów do zamówienia wraz z ich ilością
- Istnieje możliwość konfiguracji wysyłki
- Istnieje możliwość dokonania płatności
- Istnieje możliwość przejrzenia szczegółów zamówienia przed jego potwierdzeniem

Powyższe User Story jest bardzo ogólne. Obejmuje szeroki zakres funkcjonalności. Możemy je podzielić według kroków procesu, tak gdzie to zasadne, definiując bardziej precyzyjne kryteria akceptacji.

Po podziale

Jako klient chcę przeglądać dostępne w sklepie produkty, aby zapoznać się z ofertą sklepu:

- Istnieje możliwość przeglądania produktów dostępnych w sklepie
- Dostępna jest informacja o cenie każdego z produktów

Jako klient chcę wskazać produkty, aby złożyć zamówienie:

- Istnieje możliwość wskazania zamawianych produktów
- Istnieje możliwość określania ilości zamawianych produktów
- Dostępna jest informacja o aktualnej wartości zamówienia

Jako klient chcę wybrać opcję płatności, aby zapłacić w preferowany przeze mnie sposób.

- Istnieje możliwość wyboru metody płatności spośród zdefiniowanych
- Dostępne są przynajmniej opcje zapłaty przelewem tradycyjnym i przelewem elektronicznym
- Dostępna jest informacja o wybranej formie płatności

Jako klient chcę skonfigurować wysyłkę, aby otrzymać produkty w preferowany przeze mnie sposób.

- Istnieje możliwość wprowadzenia adresu dostawy
- Istnieje możliwość wyboru opcji dostawy spośród dostępnych
- Dostępna jest informacja o kosztach związanych z każdą dostępną opcją dostawy

Jako klient chcę przejrzeć zamówienie, aby zweryfikować czy jest ono poprawne.

- Istnieje możliwość przeglądania szczegółów zamówienia (lista wybranych produktów oraz ich ilość, wybrana opcja płatności, informacje o dostawie)

Jako klient chcę potwierdzić i opłacić zamówienie, aby otrzymać zamówione produkty.

- Istnieje możliwość potwierdzenia zamówienia
- Istnieje możliwość dokonania płatności w wybrany sposób

Według kryteriów akceptacji

Zdarza się, że User Story zawiera wiele kryteriów akceptacji. W takim przypadku sensowne bywa jego podzielenie względem ich podzbiorów.

Pozostając w tematyce sklepu internetowego, możemy wyobrazić sobie Historyjkę Użytkownika dotyczącą komponowania zamówienia i zawierającą sporo kryteriów akceptacji. Kryteria te mogą mówić o: możliwości zamówienia wielu produktów, możliwości edytowania listy produktów dodanych "do koszyka", możliwości anulowania zamówienia (opróżnienia "koszyka"), wyświetlania aktualnej wartości "koszyka" etc. Możliwy jest oczywiście podział na kilka niezależnych User Stories bazując właśnie na wyłączaniu poszczególnych kryteriów akceptacji.

Przykład

Jako klient chcę skomponować zamówienie, aby zamówić wybrane produkty.

- W ramach jednego zamówienia istnieje możliwość zamawiania wielu produktów
- Istnieje możliwość wyświetlenie aktualnej listy zamawianych produktów
- Istnieje możliwość sprawdzenia aktualnej, łącznej wartości zamawianych produktów
- Jeśli z danym produktem związany jest rabat, dostępna jest informacja o wysokości tego rabatu
- Istnieje możliwość edytowania listy zamawianych produktów (dodawania i usuwania pojedynczych elementów)

- → Istnieje możliwość anulowania zamówienia (opróżnienia listy zamawianych produktów)

Po podziale

Jako klient chcę zarządzać pozycjami zamówienia, aby wskazać zamawiane produkty.

- → W ramach jednego zamówienia istnieje możliwość zamawiania wielu produktów
- → Istnieje możliwość wyświetlania aktualnej listy zamawianych produktów
- → Istnieje możliwość edytowania listy zamawianych produktów (dodawania i usuwania pojedynczych elementów)
- → Istnieje możliwość anulowania zamówienia (usunięcia wszystkich zamawianych pozycji)

Jako klient chcę mieć dostęp do informacji o aktualnej wartości zamówienia i rabatach, aby kontrolować i optymalizować wartość moich zakupów.

- → Istnieje możliwość sprawdzenia aktualnej, łącznej wartości zamawianych produktów
- → Jeśli z danym produktem związany jest aktualnie rabat, dostępna jest informacja o wysokości tego rabatu

Według zakresu danych

Podział według zakresu danych jest możliwy w przypadku User Stories mówiących o wyświetlaniu lub przetwarzaniu skomplikowanych struktur lub dużych ilości danych. Weźmy pod uwagę wyświetlanie profilu użytkownika w sklepie internetowym. Profil taki może zawierać: nazwę użytkownika, podstawowe dane (imię i nazwisko, adres e-mail, telefon), wiele adresów do wysyłki, dane do wystawiania faktur, preferowaną formę płatności, preferowany sposób dostawy etc.

Jeśli User Story dotyczące wyświetlania (np. na interfejsie użytkownika) wszystkich wymienionych danych jest zbyt duże, możemy je pogrupować i stworzyć zbiór oddzielnych User Stories odpowiadających poszczególnym grupom danych.

Przykład

Jako klient chcę przeglądać swój profil użytkownika, aby sprawdzać aktualność i poprawność zawartych w nim danych.

- → Istnieje możliwość wyświetlenia nazwy użytkownika
- → Istnieje możliwość wyświetlenia danych kontaktowych użytkownika
- → Istnieje możliwość wyświetlenia listy adresów użytkownika
- → Istnieje możliwość wyświetlenia danych rozliczeniowych użytkownika
- → Istnieje możliwość wyświetlenia preferowanego przez użytkownika sposobu wysyłki
- → Istnieje możliwość wyświetlenia preferowanego przez użytkownika sposobu dostawy

Po podziale

Jako klient chcę przeglądać dane kontaktowe w profil użytkownika, aby sprawdzać aktualność i poprawność zawartych w nim danych.

- → Istnieje możliwość wyświetlenia nazwy użytkownika
- → Istnieje możliwość wyświetlenia danych kontaktowych użytkownika

Jako klient chcę przeglądać dane adresowe i rozliczeniowe w profilu użytkownika, aby sprawdzać aktualność i poprawność zawartych w nim danych.

- → Istnieje możliwość wyświetlenia listy adresów użytkownika
- → Istnieje możliwość wyświetlenia danych rozliczeniowych użytkownika

Jako klient chcę przeglądać skonfigurowane preferencje, aby sprawdzać aktualność i poprawność zawartych w nich danych.

- → Istnieje możliwość wyświetlenia preferowanego przez użytkownika sposobu wysyłki
- → Istnieje możliwość wyświetlenia preferowanego przez użytkownika sposobu dostawy

Według operacji

Typowym zbiorem operacji wykonywanych na danych jest tzw. CRUD (ang. Create, Read, Update, Delete). Jeśli zestaw operacji, wykonywanych na danych obejmuje CRUD (lub jego podzbiór), możemy rozdzielić poszczególne operacje na osobne Historyjki Użytkownika. Jeśli np. użytkownik sklepu internetowego może podawać adres dostawy (lub wiele adresów), możliwy jest podział na osobne User Stories obejmujące tworzenie adresów i ich wyświetlanie, usuwanie oraz edytowanie.

Przykład

Jako klient chcę zarządzać możliwymi adresami dostawy, aby zamawiać produkty na adres domowy, adres służbowy lub na adresy mojej rodziny i znajomych.

- → Istnieje możliwość powiązania z kontem użytkownika wielu możliwych adresów dostawy
- → Istnieje możliwość wyświetlania zdefiniowanej listy adresów dostawy
- → Istnieje możliwość dodawania nowych adresów dostawy
- → Istnieje możliwość usuwania dodanych adresów dostawy
- → Istnieje możliwość modyfikowania dodanych adresów dostawy

Po podziale

Jako klient chcę dodać wiele możliwych adresów dostawy, aby zamawiać produkty na adres domowy, adres służbowy lub na adresy dla mojej rodziny i znajomych.

- → Istnieje możliwość powiązania z kontem użytkownika wielu możliwych adresów dostawy
- → Istnieje możliwość wyświetlania zdefiniowanej listy adresów dostawy
- → Istnieje możliwość dodawania nowych adresów dostawy

Jako klient chcę zarządzać dodanymi adresami dostawy, aby móc zareagować w sytuacji, gdy przestają one być aktualne.

- → Istnieje możliwość usuwania dodanych adresów dostawy
- → Istnieje możliwość modyfikowania dodanych adresów dostawy

Według wymagań niefunkcjonalnych

Wymagania niefunkcjonalne wydają się być szczególnie "wdzięczne" pod względem podziału User Story na ich podstawie. Można tu podać wiele potencjalnych przykładów. Jednym z nich mogą być wymagania dotyczące wydajności. Załóżmy, że mamy wymaganie, iż raport ma się generować w mniej niż 5 sekund. Oczywiście, przy określonej ilości danych i określonych warunkach. W takiej sytuacji możemy opisać proces generowania raportu w ramach jednego User Story, a jego optymalizację w oddzielnym.

Innym przykładem może być dostępność interfejsu użytkownika w wielu językach, bądź możliwość korzystania z systemu na wielu platformach (np. Android i iOS). Każdy język interfejsu i obsługę na każdej z platform możemy opisać za pomocą oddzielnego User Story.

Uwaga: Wyłączanie wymagań niefunkcjonalnych do osobnego User Story bywa jednak ryzykowne. Postępując w ten sposób zwiększamy prawdopodobieństwo, że nie zostaną one nigdy zaimplementowane.

Przykład

Jako menadżer sklepu chcę wygenerować raport dotyczący zamówień z wybranego okresu, aby móc zoptymalizować proces zamawiania i magazynowania produktów.

- → Istnieje możliwość wygenerowania raportu dotyczącego zamówień, zawierającego
- → następujące dane: numer katalogowy produktu, nazwa produktu, liczba sprzedanych
- → sztuk, jednostka miary
- → Istnieje możliwość zdefiniowania zakresu dat, którego dotyczyć ma raport
- → Raport generowany jest w czasie nie dłuższym niż 5 sekund

Po podziale

Jako menadżer sklepu chcę wygenerować raport dotyczący zamówień z wybranego okresu, aby móc zoptymalizować proces zamawiania i magazynowania produktów.

- → Istnieje możliwość wygenerowania raportu dotyczącego zamówień, zawierającego następujące dane: numer katalogowy produktu, nazwa produktu, liczba sprzedanych sztuk, jednostka miary
- → Istnieje możliwość zdefiniowania zakresu dat, którego dotyczyć ma raport

Jako menadżer sklepu chcę zmniejszyć czas generowania raportu dotyczącego zamówień z wybranego okresu, po to aby jego czas jego generowania nie obniżał efektywności mojej pracy.

- → Raport generowany jest w czasie nie dłuższym niż 5 sekund

Oczywiście pomysłów na podział Historyjek Użytkownika może być dużo więcej. Przedstawione powyżej to jedynie przykłady, które nie wyczerpują pełnej listy możliwości.

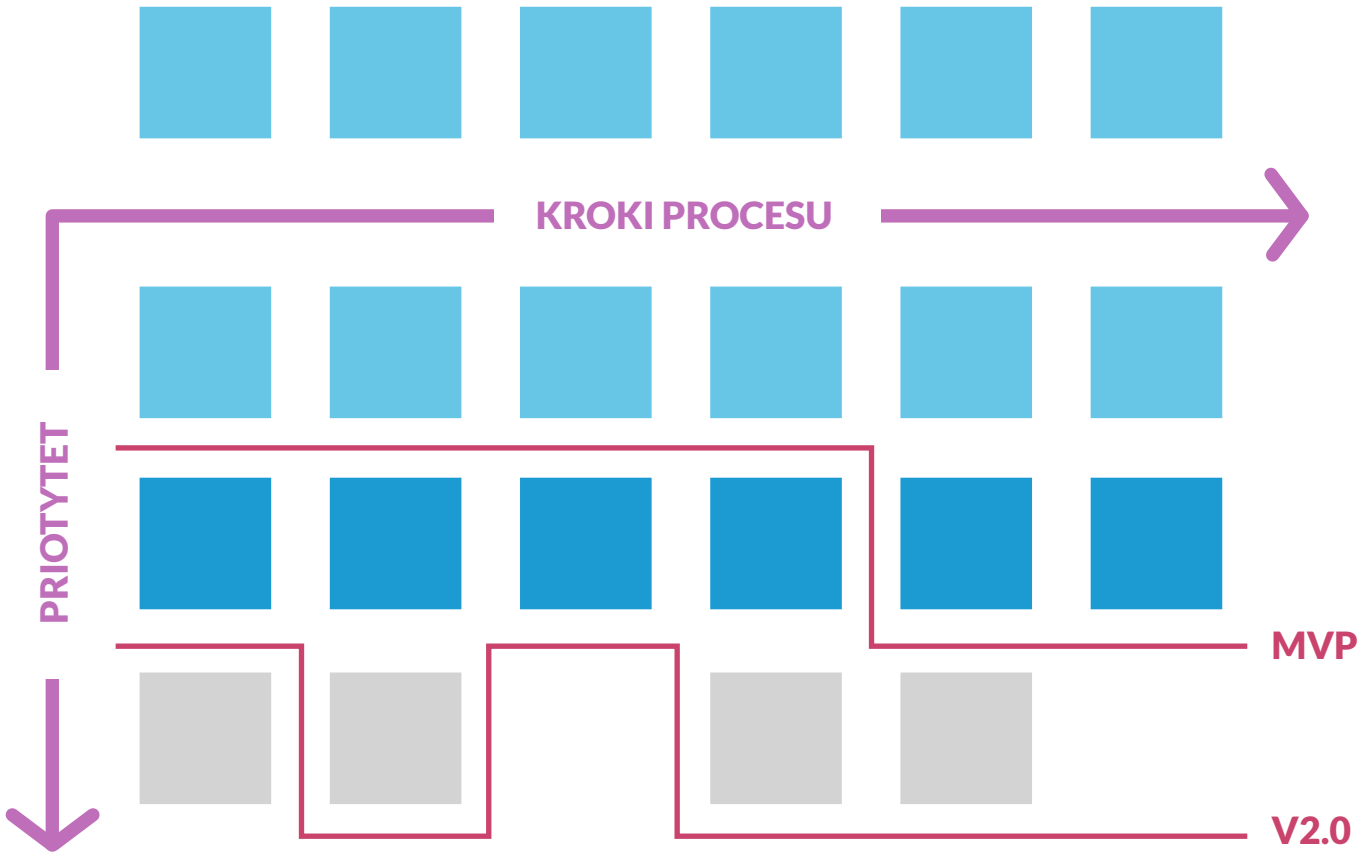
Story Mapping

Istotną wadą User Story jest to, że nie dostarcza zbyt szerokiego kontekstu. Skupia się na wybranym wymaganiu. Opisuje oczekiwania określonych interesariuszy i wartość z nimi związaną. Niestety jest to zwykle zbyt mało informacji, by zespoły deweloperskie mogły efektywnie pracować, a na pewno za mało, aby móc w pełni zrozumieć ogólny cel projektu i potrzebę, która za nim stoi. **Z tego właśnie powodu opisywanie wymagań w formie User Stories warto wesprzeć innymi technikami i formami dokumentacji.** Jedną z takich technik jest Story Mapping.

Story Mapping – co to takiego?

W wielkim skrócie jest to mapowanie User Stories (lub opcjonalnie wymagań w innej formie) na kroki procesu. Musimy zatem określić proces, jego poszczególne kroki i przypisać im określone Historyjki Użytkownika. Mamy też możliwość określania priorytetów poszczególnych User Stories oraz możemy w wygodny sposób wyznaczać zakres poszczególnych wersji produktu.

Schemat Mapy Historyjek

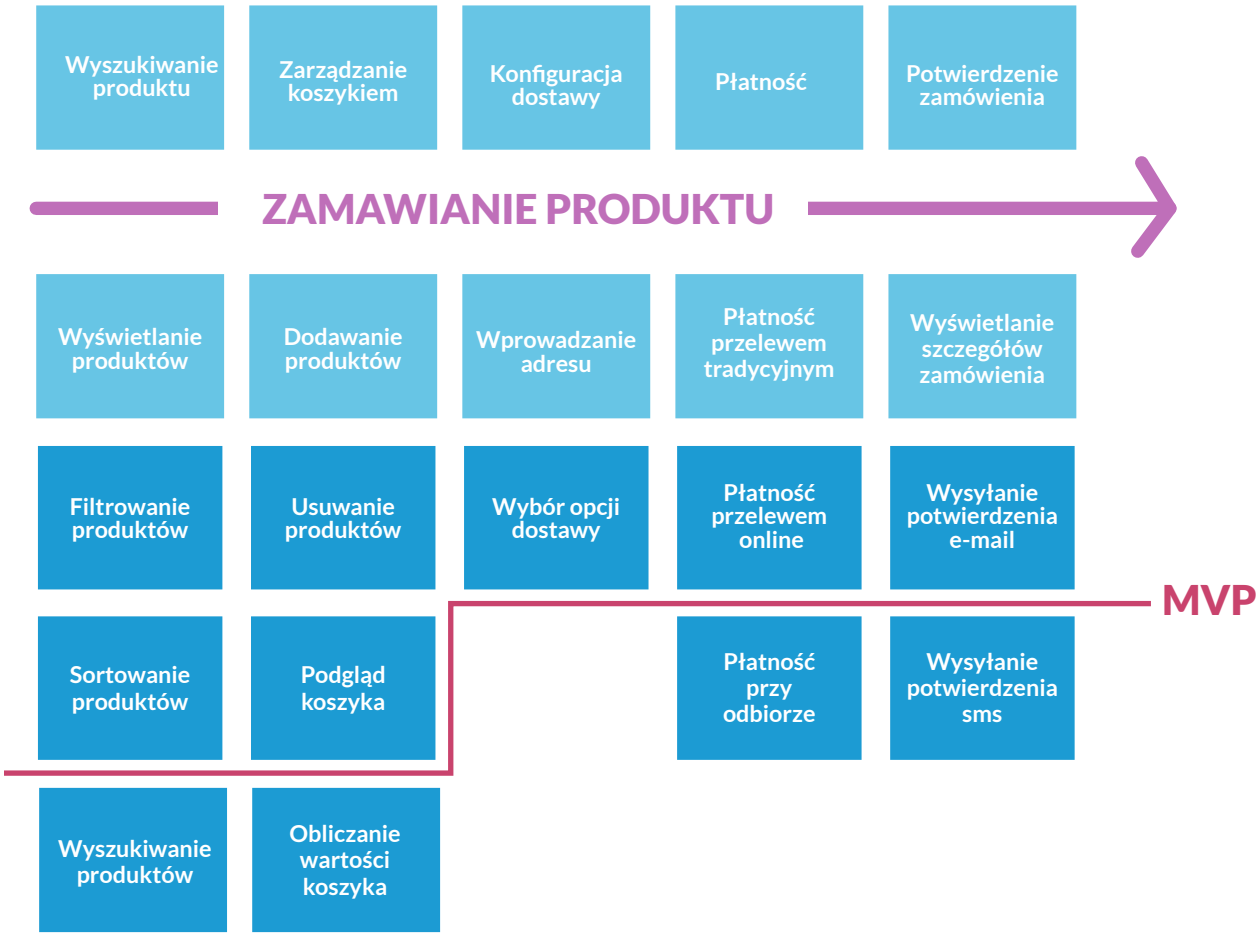


Na powyższym schemacie widzimy dwie osie: poziomą i pionową. Oś pozioma wyznacza przebieg procesu. Nad nią pojawiają się poszczególne kroki procesu, od lewej do prawej, układające się w spójną całość. Oś pionowa wskazuje priorytet. Im niżej dana Historyjka Użytkownika jest umieszczona, tym niższy jest jej priorytet. Czerwone linie wyznaczają zakres poszczególnych wersji produktu. MVP (ang. Minimal Viable Product), to minimalny, biznesowo uzasadniony zakres, dostarczający użytkownikom wartości. Kolejna linia, wyznacza zakres, który będzie dodany w następnej wersji.

Story Mapping – konkretny przykład

Rozważmy konkretny przykład zastosowania Story Mappingu dla procesu zamawiania produktów w sklepie internetowym. Przykład jest banalny, co pozwala skupić się na prezentacji techniki, a nie zawiłościach danego procesu.

Oczywiście tworząc Mapę Historyjek, warto jest używać tytułów User Story, a nie pełnego Statement of Value.



Przykładowy proces zamawiania produktu w sklepie internetowym składa się z kilku, następujących po sobie, kroków. Widzimy je nad poziomą osią wyznaczającą kierunek procesu. Każdemu z kroków procesu zostało przyporządkowanych kilka User Stories. Zostały one ułożone według priorytetów. Te ważniejsze znajdują się wyżej, a mniej ważne niżej. Widzimy również linię wyznaczającą MVP a zatem zakres funkcjonalności (wymagań), który będzie ujęty w pierwszej, podstawowej wersji. Oczywiście potencjalnie moglibyśmy zaplanować kolejne wersje i zawrzeć w nich pozostałe wymagania.

Story Mapping w formie warsztatu

Główną zaletą Story Mappingu jest to, że pozwala przekazać i omówić szerszy kontekst. Nie skupiamy się na perspektywie pojedynczego User Story, ale widzimy całość procesu lub nawet całość rozwiązania, które tworzymy. W ramach tej całości, możemy umiejscowić poszczególne User Stories. Dzięki temu możliwe jest zastosowanie podejścia "od ogółu do szczegółu". **Podejście takie wydaje się być naj właściwsze, ponieważ jedynie mając szeroką perspektywę i zrozumienie celu oraz potrzeby biznesowej, zespoły deweloperskie są w stanie opracować optymalne rozwiązanie.**

Mapa Historyjek (ang. Story Map) może być oczywiście formą dokumentacji. Mamy na niej opisane ogólne kroki procesu, odpowiadające im User Stories i ich priorytety. Jest to więc opcja, którą możemy rozważyć jako dokumentację naszego rozwiązania. Moim zdaniem nie jest to jednak optymalne zastosowanie techniki Story Mappingu. Największą zaletą i optymalnym zastosowaniem tej techniki jest możliwość osiągnięcia wspólnego zrozumienia pomiędzy interesariuszami. Zrozumienie to może dotyczyć potrzeby biznesowej, odpowiadającego jej rozwiązania oraz jego zakresu.

Organizując warsztat, którego celem jest Story Mapping i zapraszając odpowiednich interesariuszy (zarówno "biznesowych" jak i "technicznych" zyskujemy świetną okazję do wymiany informacji, dyskusji i osiągania bardzo pożądanego zjawiska, jakim jest wspomniane wspólne zrozumienie. Jest to niezwykle ważne w kontekście precyzyjnej estymacji rozwiązania i efektywnej współpracy nad jego tworzeniem. Tworzenie Mapy Historyjek to również dobra okazja do dyskutowania priorytetów i podziału na wersje.

Wady Story Mappingu

Wiemy już, jakie są zalety tworzenia Map Historyjek. Oczywiście jak każda inna, technika ta również posiada pewne wady i ograniczenia. Przede wszystkim ma zastosowanie głównie tam, gdzie występują procesy. Polega w końcu na mapowaniu wymagań na

poszczególne ich kroki. W przypadku systemów, w których niemożliwe jest zdefiniowanie jasnych procesów, wciąż możemy starać się używać Story Mappingu, grupując wymagania według pewnych cech, czy funkcjonalności. Jest to jednak mniej wygodne i technika staje się mniej przydatna. W przypadku braku procesu, pozioma oś mapy traci swój sens.

Kolejne ograniczenie omawianej techniki jest również związane z procesami. W przypadku bardzo skomplikowanych i rozgałęzionych procesów, określenie Story Map bywa trudne. Należy wtedy, w ramach techniki Story Mappingu, zastosować uproszczoną wersję procesu, a szczegółową i dokładną przedstawić w innej formie (np. odpowiedniego diagramu).

Ostatnim ograniczeniem, jakie warto wspomnieć jest brak informacji o wzajemnych powiązaniach pomiędzy wymaganiami. Niektóre wymagania muszą być zrealizowane przed innymi z przyczyn technicznych lub biznesowych. Czasami też implementacja jednego wymagania może znacząco wpłynąć na koszt implementacji innego. Tego rodzaju zależności nie są uwzględniane na Mapach Historyjek i muszą być analizowane i dokumentowane w innej formie.

Podsumowanie

User Story jest użyteczną i popularną i techniką dokumentowania wymagań. Świetnie nadaje się do dokumentowania wymagań interesariuszy. **Niestety ze względu na swoje zalety, bywa często nadużywane.** W wielu przypadkach, efektywnym jest wsparcie lub zastąpienie User Story innymi technikami opisywania wymagań. W zależności od sytuacji, mogą to być różnego rodzaju modele (diagramy), makiety interfejsu użytkownika, opisy Przypadków Użycia lub opisy w języku naturalnym w innej, niż User Story, formie.

Używając User Story zawsze należy zadawać pytanie czy jest to optymalna technika do użycia w określonej sytuacji. Jeśli tak jest, warto postarać się o jak najwyższą jakość tworzonych Historyjek Użytkownika.

Referencje

Agile Extension to the BABOK® Guide. Version 2. 2017.
A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide). Version 3. 2015